

# Bayesian Network Model for Learning Arithmetic Concepts

Yali Lv<sup>a,b,\*</sup>, Tong Jing<sup>a</sup>, Yuhua Qian<sup>b</sup>, Jiye Liang<sup>b</sup>, Jianai Wu<sup>a</sup>, and Junzhong Miao<sup>a</sup>

<sup>a</sup>*School of Information Management, Shanxi University of Finance and Economics, Taiyuan, 030006, China*

<sup>b</sup>*Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, Shanxi University, Taiyuan, 030006, China*

---

## Abstract

An object usually belongs to multiple concepts, but some concepts can be judged directly while other concepts need to be inferred indirectly. To learn some arithmetic concepts from positive integer number sets, we address an arithmetic concept Bayesian network (ACBN) model by taking advantage of Bayesian networks. Specifically, we first give an ACBN model to represent the arithmetic concept knowledge and their direct relationships, and then we design an ACBN model learning algorithm based on domain knowledge. Furthermore, to infer indirectly some arithmetic concepts, we design the learning method of evidence concepts based on the idea of k-nearest neighbors, and then we propose the inference algorithm of the ACBN model. Finally, the experimental results demonstrate that the ACBN model can effectively learn some daily arithmetic concepts.

**Keywords:** probabilistic inference; Bayesian networks; arithmetic concepts; evidence concepts; domain knowledge

(Submitted on November 14, 2018; Revised on December 15, 2018; Accepted on January 13, 2019)

© 2019 Totem Publisher, Inc. All rights reserved.

---

## 1. Introduction

As a probabilistic graph model, Bayesian network (BN) [1-2] has become an important tool to represent and infer uncertainty knowledge. It can simulate the human brain's reasoning process based on its representation of dependence and independence relationships among variables. Additionally, many efficient BN model learning algorithms [2-4] and inference algorithms [2, 5-6] have been proposed. More and more research fields have begun to use the BN model to explore the structure of the problem and further solve it.

Recently, human level concept learning and reasoning have made great progress. In 2015, Lake et al. [7] proposed a computational model that captures human learning ability in a simple visual concept: handwritten characters from the world's alphabets. This model reached human level in one classification task, simultaneously defeating the deep learning methods. George et al. [8] proposed a generative vision model that breaks the defence of modern text-based CAPTCHAs. Its learning and reasoning outperform deep neural network learning. Some motion recognition research [9] is also conducted based on the ideas of concept learning. However, most of the existing works focus on concept learning and pattern recognition in the visual field. For arithmetic concept learning, Tenenbaum researched Bayesian concept learning by taking "number game" as an example in [10]. The research explains and simulates the general way of humans' thinking about concept learning by a computer. However, there is a lack of consideration of the relationships among arithmetic concepts; thus, to make it possible for a machine to simulate human reasoning in the face of some concepts, it is urgent to study a concept representation framework that can reflect the relationships among arithmetic concepts.

The BN model is the combination of graph theory and probability theory. Since it has unique advantages in knowledge representation and knowledge reasoning, we can infer and learn the hidden knowledge from observed data by constructing BN's structure and defining the probability distribution regardless of where data come from and how complex they are. Thus, in this paper, we address an arithmetic concept Bayesian network to learn and infer the relationships among arithmetic concepts, which are list in [10].

---

\* Corresponding author.

E-mail address: [xlvyali@126.com](mailto:xlvyali@126.com)

The rest of this paper is arranged as follows. Section 2 briefly reviews some basic contents about BN. In Section 3, we give the definition of the arithmetic concept Bayesian network (ACBN) and then study the learn algorithm of the ACBN model. Furthermore, we propose the learning method of evidence concepts and the inference algorithm of the ACBN model. The experimental results demonstrate the performance of the ACBN model in Section 4. Finally, Section 5 concludes the paper.

## 2. Bayesian Network Model

In order to address a Bayesian network for learning arithmetical concepts, we provide a necessary introduction on the Bayesian network as well as its learning and inference algorithms.

The BN model encodes a complex distribution over a high-dimensional space [11] based on directed graph representation by dependence and independence among variables. It is mainly composed of directed acyclic graph (DAG) structure and each node's conditional probabilistic distribution (CPD). That is, a BN  $\mathcal{M}=(\mathcal{G},\theta)$  comprises a DAG structure  $\mathcal{G}=(V,E)$  and the set of each variable's CPDs  $\theta$ , where

- $V=\{X_1,X_2,\cdots,X_n\}$  denotes the variable set.
- $E=\{<X_i,X_j>|X_i\neq X_j\}$  is the set of edges reflecting the direct probabilistic interactions among parent- child variables.
- $\theta=\{p(X_i|\pi(X_i))\}$  is the set of CPDs reflecting the probabilistic dependence degree among parent-child variables, and  $\pi(X_i)$  is the parent set of variable  $X_i$ .

The joint probabilistic distribution is

$$p(X_1,X_2,\cdots,X_n)=\prod_i p(X_i|\pi(X_i)) \quad (1)$$

This means, for any variable  $X_i$ , given the set of parent nodes  $\pi(X_i)$ ,  $X_i$  is independent of all non-descendant nodes except for  $\pi(X_i)$ .

For BN model learning from data, the method based on score and search includes two aspects: model selection and model optimization.

(1) The criteria of model selection are also called the scoring function, and the function evaluates the fitting degree between the learned model  $\mathcal{M}$  and the observed data  $\mathcal{D}$ . Here, we only introduce the BIC (Bayesian information criterion) scoring function [12-13], and its form is

$$\text{BIC}(\mathcal{D}|\mathcal{M})\approx\log p(\mathcal{D}|\mathcal{G},\theta^*)-\frac{d}{2}\log(m) \quad (2)$$

Where  $\theta^*$  is the maximum likelihood estimation (MLE) of parameter  $\theta$  given a DAG  $\mathcal{G}$ ,  $d$  denotes the number of independent parameters in a model, and  $m$  is the size of sample data. The function shows that the higher the score, the better the learned network.

(2) For BN model optimization, there are many famous algorithms, such as K2, hill climbing, and MCMC. In this paper, we optimize the BN model based on hill climbing and domain knowledge.

After learning the BN model, we aim at probabilistic inference or using the distribution to answer some queries. That is, we need to design some algorithms that can be used to compute the posterior probability of some variables. There are many classic inference algorithms, such as variable elimination, clique tree, and variational inference. Here, we mainly discuss the evidence's learning and directly use the clique tree algorithm to infer, so we do not introduce any inference algorithms in detail.

## 3. Arithmetic Concept Bayesian Network Model

In this section, in order to directly or indirectly learn the arithmetic concepts and indirectly infer concept relationships

among arithmetic concepts, we address the arithmetic concept Bayesian network (ACBN) model. First, we give the problem statement of ACBN for learning and inferring arithmetic concepts, and then we design the learning algorithm of the ACBN model from data. Furthermore, the inference algorithm and the learning algorithm of evidence concepts are studied.

### 3.1. Problem Statement

The arithmetic concepts of “number game” in [10] mainly include “even”, “squares”, “multiples of 5”, “ends in 6”, “prime”, “powers of 3”, and so on. Given a positive integer number, such as “8”, we know that it belongs to the concepts “even”, “multiples of 2”, “powers of 2”, and so on. After having observed a group of numbers, such as  $\{4, 16\}$ ,  $\{9, 26, 36\}$ , and  $\{2, 8, 32, 64, 72\}$ , we will judge which concepts they belong to. Therefore, we need some methods to learn them directly or indirectly.

To judge whether an example belongs to a concept  $C$ , we can select randomly some positive examples of  $\mathcal{D} = \{x_1, x_2, \dots, x_n\}$  that belong to the concept  $C$ , and then we can determine whether a new test set  $\tilde{x}$  is the concept  $C$  by observing these positive examples. This is equivalent to binary classification in machine learning. To see this, we can define the function as follows.

$$f = \begin{cases} 1, & \text{if } x \text{ is an example of concept } C \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

Therefore, the purpose of this paper is to define a Bayesian network that can learn arithmetic concepts and infer among arithmetic concepts. The arithmetic concepts are all binary, and their values are all 1 or 0 as the indicator function  $f$  in Equation (3).

Overall, there are two problems that need to be solved.

(1) Learning the BN model of arithmetic concepts from observed data: in this step, since learning the BN model is an NP-hard problem, the time complexity increases exponentially with the number of variables. We need to consider how to limit BN's search space to decrease time complexity.

(2) Inferring among arithmetic concepts based on ACBN model: in this step, since test samples are not different from train samples, we first need to learn exact evidence concepts before inferring other concepts, that is, the evidence concepts need to be obtained before computing the posterior probability.

### 3.2. ACBN Model Learning

#### 3.2.1. Definition of ACBN

Based on the formalized representation of BN, we define the arithmetic concepts Bayesian network (ACBN) to represent the direct probabilistic relationships among arithmetic concepts and corresponding uncertainties.

**Definition 1** (Arithmetic concept Bayesian network, ACBN) An arithmetic concept Bayesian network, abbreviated as ACBN, is a pair  $(\mathcal{G}_{AC}, \theta)$ , where

- $\mathcal{G}_{AC} = (AC, E)$  is the DAG of the ACBN model and  $AC$  is the set of arithmetic concept nodes in  $\mathcal{G}_{AC}$ . Each node in  $AC$  corresponds to an arithmetic concept  $X_i$ , and its value is 1 or 0, where 1 indicates that  $X_i$  belongs to the concept and 0 indicates that  $X_i$  does not belong to the concept.  $E$  is the directed edge set, which denotes the direct similarity relationships between arithmetic concepts.
- $\theta = \{p(X_i | \pi(X_i))\}$  is the set of conditional probabilities, and it consists of the CPT of node  $X_i$  in  $\mathcal{G}_{AC}$ . Similarly,  $\pi(X_i)$  is the set of parent nodes of arithmetic concept node  $X_i$ .

Similarly, the joint probabilistic distribution of the ACBN model is shown in Equation (1).

Furthermore, to learn the ACBN in Definition 1 and infer among arithmetic concepts, we will address the graphical

structure, learn evidence concepts, and design the inference algorithm, which will be studied in this subsection.

### 3.2.2. Learning Algorithm

As mentioned in Section 2, we adopt the BIC score to select the BN model and use the hill climbing strategy to optimize the BN space, but the hill climbing algorithm excessively depends on the given initial DAG, and it is easy to get into the local optimal solution. Here, in order to avoid this disadvantage, we give the initial DAG based on domain knowledge (DK) that easily reflects the relationships between arithmetic concepts.

For example, if a number or a group of numbers belongs to “odd (  $A$  )”, it does not belong to “even (  $B$  )”, that is, there is no edge  $\langle A, B \rangle$  in BN's DAG; if a number or a group of numbers is “ends in 1 (  $K$  )”, it cannot be “ends in is 2 (  $L$  )”, and there is also no edge  $\langle K, L \rangle$ . In the same way, if a number or a group of numbers is “multiples of 6 (  $G$  )”, it must be an “even (  $B$  )”, and there must be an edge  $\langle G, B \rangle$  in BN's DAG; if a number or a group of numbers is “powers of 2 (  $T$  )”, it must be an “even (  $B$  )”, there must also be an edge  $\langle T, G \rangle$ , and so on. They are all regarded as domain knowledge (DK).

Therefore, when we optimize the initial network, the initial DAG should contain such domain knowledge that will be used to limit BN's search space. These edges that should not exist must not be added in the process of climbing; these edges that should exist should not be subtracted in the process of climbing as much as possible. That is, only these edges that cannot be given based on the domain knowledge can be used as the edges to be selected in the candidate network space.

The goal of the ACBN learning algorithm is to give the initial BN model structure based on domain knowledge DK and then to obtain its candidate DAGs by adding edge  $O_{\text{add}}$ , subtracting edge  $O_{\text{sub}}$ , or reversing edge  $O_{\text{rev}}$ . Note that the three operators of searching candidate network structures are different from the basic hill climbing strategy. (1) We can use  $O_{\text{add}}$  to add a directed edge to the current network structure; (2)  $O_{\text{sub}}$  means to subtract a directed edge from the current network structure, except the edges in DK; (3)  $O_{\text{rev}}$  denotes reversing the direction of an edge in the current network structure, except the edges in DK. Moreover, the preconditions of  $O_{\text{add}}$  and  $O_{\text{rev}}$  should still be that there is no cycle in the current network, and the network is still a DAG.

Furthermore, the candidate network structure is optimized according to the BIC score criterion and hill climbing strategy. The parameters of the ACBN model are learned by using maximum likelihood estimation (MLE). The pseudo-code of the ACBN learning algorithm is described by Algorithm 1.

---

**Algorithm 1** ACBN learning algorithm

---

**Input:** Arithmetic concept variables  $X$ ; complete data  $\mathcal{D}$ ; BIC score function; domain knowledge DK;

**Output:** ACBN model  $(\mathcal{G}, \theta) = (V, E, \theta)$

---

```

1:  $V \leftarrow X$ ; Given  $E$  based on DK;
2:  $\mathcal{G}^0 \leftarrow (V, E)$ ;  $\mathcal{G} \leftarrow \mathcal{G}^0$ ;
3:  $\theta \leftarrow$  parameters' MLE of  $\mathcal{G}$ ;  $\text{BIC}_{\text{old}} \leftarrow \text{BIC}(\mathcal{G}, \theta | \mathcal{D})$ ;
4: while true do
5:    $\mathcal{G}^* \leftarrow \text{null}$ ;  $\theta^* \leftarrow \text{null}$ ;  $\text{BIC}_{\text{new}} \leftarrow -\infty$ ;
6:   for  $\mathcal{G}'$  that obtained by using each operator in Section 3.2.2 based on DK once for  $\mathcal{G}$  do
7:      $\theta' \leftarrow$  parameters' MLE of  $\mathcal{G}'$ ;  $\text{BIC}_{\text{temp}} \leftarrow \text{BIC}(\mathcal{G}', \theta' | \mathcal{D})$ ;
8:     if  $\text{BIC}_{\text{temp}} > \text{BIC}_{\text{new}}$  then
9:        $\mathcal{G}^* \leftarrow \mathcal{G}'$ ,  $\theta^* \leftarrow \theta'$ ,  $\text{BIC}_{\text{new}} \leftarrow \text{BIC}_{\text{temp}}$ ;
10:    end if
11:  end for
12:  if  $\text{BIC}_{\text{new}} > \text{BIC}_{\text{old}}$  then
13:     $\mathcal{G} \leftarrow \mathcal{G}^*$ ,  $\theta \leftarrow \theta^*$ ,  $\text{BIC}_{\text{old}} \leftarrow \text{BIC}_{\text{new}}$ ;
14:  else
15:    return  $(\mathcal{G}, \theta)$ ;
16:  end if
17: end while

```

---

### 3.3. ACBN Model Inference

This section mainly studies the inference algorithm among arithmetic concepts based on the ACBN model. We first discuss

how to learn evidence concepts and then give the inference algorithm.

### 3.3.1. Evidence Concept Learning based on kNN

When inferring among arithmetic concepts based on the ACBN model, we first need to get the evidence concepts. However, for the arithmetic concept test data set, since each test data  $\tilde{x}$  contains not only a number but also a group of numbers, how do we decide which concepts they belong to?

The  $k$  nearest neighbors (kNN) algorithm is a classic algorithm for classification in machine learning. The idea of the kNN algorithm is to look for  $k$  nearest sample for each test data  $\tilde{x}$  and then judge which class most of its neighbor samples belong to, so we have that  $\tilde{x}$  belongs to the class. In this paper, this idea is applied to the judgement of evidence concepts in the ACBN model, but there are two differences from the basic kNN algorithm.

(1)  $k$  is not fixed in each test sample data  $\tilde{x}$ , which is finally determined by the element number of  $\tilde{x}$ . That is,  $k = |\tilde{x}|$ , where  $|\cdot|$  denotes the element number. For example, if the test sample is  $\{45, 80\}$ , we have  $k = 2$ ; if the test sample is  $\{4, 68, 67, 89, 80\}$ , we have  $k = 5$ .

(2) Generally, the kNN algorithm claims that most of the samples in the nearest samples belong to the same class, and we will determine that the test sample also belongs to this class. However, in our method, we ask for that all samples belong to the same class, and we can determine that the test sample belongs to this class. The reason is that the purpose of our work is to obtain the evidence concepts, which must be absolutely certain evidence, and the values must be determined exactly, not in the form of probabilities to be determined.

Furthermore, we need to consider the distance metric of kNN. In this paper, each element in a group of numbers is its neighbors. First, assuming  $\tilde{x}$  is a test sample data, we denote its  $k$  neighbors by  $Nb(\tilde{x})$  and then count the concept class in  $Nb(\tilde{x})$ . We then take the concepts all belonging to the same concept or not belonging to the same concept in  $k$  neighbors as the evidence concepts. In general, we can obtain many evidence concepts because they do not all belong to the same concepts in  $k$  neighbors. We usually set the number of evidence concepts  $N = s$ . For example, supposing  $\tilde{x} = \{6, 18, 24\}$ , according to  $k$  nearest neighbors, we find three nearest neighbors  $\{6\}$ ,  $\{18\}$ , and  $\{24\}$ , so we use them to judge that they all belong to “even” ( $B = 1$ ), “multiples of 6” ( $G = 1$ ), “not ends in 3” ( $O = 0$ ), and so on. If  $s = 2$ , we get two positive evidence concepts  $B = 1, G = 1$  as far as possible.

### 3.3.2. Inference Algorithm

Given a group of numbers  $\tilde{x}$ , we need to learn which concepts  $C(\tilde{x})$  they belong to. We can learn some direct arithmetic concepts by getting evidence concepts. In addition, we can learn some indirect arithmetic concepts by ACBN model inference. In this paper, we use the classic joint tree inference algorithm to infer other arithmetic concepts, which predicts all concept classes in the form of probabilities, unlike other algorithms that predict a concept class. In conclusion, the main idea is to get evidence arithmetic concepts based on kNN and then infer all other arithmetic concepts by the joint tree inference method in BN. Here, we only get the top  $t$  arithmetic concepts by their values of probability. The pseudo-code of the ACBN inference algorithm is shown as Algorithm 2.

---

#### Algorithm 2. ACBN inference algorithm

---

**Input:** Arithmetic concepts  $C(C \in \{0, 1\}^n)$ ; test data set  $\tilde{D} = \{\tilde{x}_m\}$ ;

**Output:** All top  $t$  arithmetic concepts

1: Constructing ACBN model  $\mathcal{M}_{ACBN}$  by Algorithm 1;

2: **for**  $i = 1$  to  $m$  in  $\tilde{D}$  **do**

3:     finding  $|D_i|$  nearest neighbors;

4:     All evidence concepts  $\leftarrow$  arithmetic concept set of  $Nb(D_i)$ ;

5:     Evidence concepts  $\leftarrow$  select  $s$  in all evidence concepts;

6:     Get top  $t$  arithmetic concepts based on joint tree inference algorithm except that evidence concepts;

7: **end for**

---

## 4. Experiments

In this section, we will analyze the concept learning performance of the ACBN model by comparing with the order of exact frequency. It is divided into two parts to verify in the experiments: (1) we analyze the learning results of the ACBN model that reflect the relationships among arithmetic concepts, especially for different initial BNs based on domain knowledge; (2) based on the learned ACBN model, we verify the inference results of arithmetic concepts by comparing with exact frequency and explaining human cognition.

### 4.1. Experiment Setting and Data Set

The experiments have been implemented in R language environment by using a *bnlearn* toolbox [14]. The iteration number in hill climbing is set to 5000; the edge number range of given domain knowledge is from 10 to 37 in the initial networks; we use the BIC score shown in Equation (2) as the scoring function; the number range of evidence concepts for ACBN inference is from 2 to 5. The data set is positive integer numbers within 100. There are 30 arithmetic concepts that are shown in Table 1, and each arithmetic concept is written as  $A \sim Z, a, b, c, d$ , respectively.

Table 1. 30 arithmetic concepts

No.	Concepts
1-5	odd ( $A$ ), even ( $B$ ), squares ( $C$ ), multiples of 3 ( $D$ ), multiples of 4 ( $E$ )
6-9	multiples of 5 ( $F$ ), multiples of 6 ( $G$ ), multiples of 7 ( $H$ ), multiples of 8 ( $I$ )
10-14	multiples of 9 ( $J$ ), ends in 1 ( $K$ ), ends in 2 ( $L$ ), ends in 3 ( $M$ ), ends in 4 ( $N$ )
15-19	ends in 5 ( $O$ ), ends in 6 ( $P$ ), ends in 7 ( $Q$ ), ends in 8 ( $R$ ), ends in 9 ( $S$ )
20-24	powers of 2 ( $T$ ), powers of 3 ( $U$ ), powers of 4 ( $V$ ), powers of 5 ( $W$ ), powers of 6 ( $X$ )
25-28	powers of 7 ( $Y$ ), powers of 8 ( $Z$ ), powers of 9 ( $a$ ), powers of 10 ( $b$ )
29-30	powers of 3-32 ( $c$ ), prime number ( $d$ )

As for the construction of the training sample data, each number within 100 is taken as a sample data, and they are shown in Table 2. It is a  $100 \times 30$  two-dimensional table. Arbitrary numbers greater than 1 and within 100 are used as test sample data, and they are shown in Table 3.

Table 2. Arithmetic concept training data sets within 100

	A	B	C	D	E	F	G	H	.....	X	Y	Z	a	b	c	d
{1}	1	0	1	0	0	0	0	0	.....	0	0	0	0	0	0	0
{2}	0	1	0	0	0	0	0	0	.....	0	0	0	0	0	1	1
.....									.....							
{100}	0	1	0	0	1	1	0	0	.....	0	0	0	0	1	0	0

Table 3. Arithmetic concept testing data sets within 100

	A	B	C	D	E	F	G	H	.....	X	Y	Z	a	b	c	d
{13, 23}	1	0	0	0	0	0	0	0	.....	0	0	0	0	0	0	1
.....									.....							
{4, 60, 66}	0	1	0	0	0	0	0	0	.....	0	0	0	0	0	0	0
.....									.....							
{9, 19, 39, 89, 99}	1	0	0	0	0	0	0	0	.....	0	0	0	0	0	0	0
.....									.....							
{1, 2, 3, 4, 5, ....., 99, 100}	0	0	0	0	0	0	0	0	.....	0	0	0	0	0	0	0

### 4.2. ACBN Learning Results and Analysis

In this subsection, we learn the ACBN model by Algorithm 1 and then analyze the ACBN learning results. As mentioned above, the initial ACBN is very important for the learning results. To learn the ACBN model better and more efficiently, we give some edges that should be had and should not be had based on domain knowledge, and the domain knowledge is shown in Table 4.

For initial ACBN structure  $\mathcal{G}^0$ , we give different initial networks having different edge numbers based on domain knowledge; they are 37, 17, and 10 in the experiments, respectively. Since the initial network that includes 37 given edges

has the most information, we give this initial network only one time to learn the ACBN model. The other initial networks that include 17 and 10 given edges are given five times to learn the ACBN model. The learned results of three cases are shown in Table 5.

Table 4. The relationships of edges in an initial BN based on domain knowledge

Not having edges	$\langle A, B \rangle$ ,	$\langle A, E \rangle$ ,	$\langle B, D \rangle$ ,	$\langle A, G \rangle$ ,	$\langle A, I \rangle$ ,	$\langle A, L \rangle$ ,
	$\langle A, N \rangle$ ,	$\langle A, P \rangle$ ,	$\langle A, R \rangle$ ,	$\langle A, T \rangle$ ,	$\langle A, V \rangle$ ,	$\langle A, X \rangle$ ,
	$\langle A, Z \rangle$ ,	$\langle A, b \rangle$ ,	$\langle A, c \rangle$ ,	$\langle B, c \rangle$ ,	$\langle K, L \rangle$ ,	$\langle K, M \rangle$ ,
	$\langle K, N \rangle$ ,	$\langle K, O \rangle$ ,	$\langle K, P \rangle$ ,	$\langle K, Q \rangle$ ,	$\langle K, R \rangle$ ,	$\langle K, S \rangle$ ,
	.....					
Having edges	$\langle A, K \rangle$ ,	$\langle A, M \rangle$ ,	$\langle A, O \rangle$ ,	$\langle A, Q \rangle$ ,	$\langle A, S \rangle$ ,	$\langle A, U \rangle$ ,
	$\langle A, W \rangle$ ,	$\langle A, Y \rangle$ ,	$\langle A, a \rangle$ ,	$\langle B, E \rangle$ ,	$\langle B, G \rangle$ ,	$\langle B, I \rangle$ ,
	$\langle B, L \rangle$ ,	$\langle B, N \rangle$ ,	$\langle B, P \rangle$ ,	$\langle B, R \rangle$ ,	$\langle B, T \rangle$ ,	$\langle B, V \rangle$ ,
	$\langle B, X \rangle$ ,	$\langle B, Z \rangle$ ,	$\langle B, b \rangle$ ,	$\langle B, c \rangle$ ,	$\langle D, G \rangle$ ,	$\langle D, J \rangle$ ,
	$\langle D, U \rangle$ ,	$\langle D, X \rangle$ ,	$\langle D, a \rangle$ ,	$\langle E, I \rangle$ ,	$\langle E, V \rangle$ ,	$\langle E, Z \rangle$ ,
	$\langle F, O \rangle$ ,	$\langle F, W \rangle$ ,	$\langle F, b \rangle$ ,	$\langle G, X \rangle$ ,	$\langle H, Y \rangle$ ,	$\langle I, Z \rangle$ ,
	$\langle J, a \rangle$ ,	.....				

Table 5. ACBN learning results

Cases	Edge number in $\mathcal{G}^0$ based on DK	Edge number in learned ACBN	BIC score in learned ACBN
1	37	46	-830.67
2	17	36	-774.28
	17	38	-782.20
	17	36	-767.77
	17	37	-781.31
	17	37	-781.20
3	10	35	-762.00
	10	34	-762.21
	10	33	-765.29
	10	<b>32</b>	<b>-756.71</b>
	10	34	-765.77

The higher the BIC score, the better the ACBN model fits the data in the same size data set. From Table 5, we have the following conclusions.

(1) The BIC score is the highest in the third case, so the learned network is the optimal one. Although the learned network has the most edges and includes richer information in the first case than the others, its BIC score is not the highest. The reason is that the BIC score function is the evaluation criteria with penalty items, which is given according to the complexity of the model. Therefore, for a model, it is not necessarily better to give more domain knowledge or to learn more edges. It is possible that the network contains a large amount of redundant information. The model should be simplified while expressing the same effect, which also fits with the principle of Occam's razor.

(2) There is no significant difference in BIC scores and the learned edge number between the second case and the third case. We have that the given optimal edge number based on domain knowledge is from 10 to 20. Thus, we choose the network with the highest BIC score, the third case, as the learned optimal ACBN model. That is, BIC = -756.71 and the edge number is 32, as shown in bolded font in Table 5. The learned ACBN model structure is shown in Figure 1.

#### 4.3. ACBN Inference Results and Analysis

In this subsection, we infer other arithmetic concepts based on the ACBN model by Algorithm 2 and then analyze the ACBN inference results. The number range of elements of sample data in data sets is within 3, and we randomly select some samples as the test sample data sets in this experiment. The evidence concepts are learned according to our proposed method, and the evidence number is at most 5 here. In addition, to make explanation easy, we choose only the first two concepts with the highest probability value. Finally, we compare and explain the inference results with the exact frequency. Test sample data and inference results are shown in Table 6. From Table 6, we have the following three conclusions.

##### 4.3.1. The Inference Results are Correct and Explicable

The inference results based on the ACBN model are consistent with the exact frequency statistics results in rank for each case, which shows that our method of probabilistic inference among arithmetic concepts is correct and feasible. For example,





the same evidence concepts have no influence on the inference result. Each case corresponds to a kind of evidence concept in Table 6, but there are several kinds of sample data that are suitable for the case. For example, for the second case in Table 6, since two kinds of test sample data  $\{37\}$  and  $\{23, 61, 89\}$  have the same evidence concepts, they have the same inference results. Thus, we focus on studying the relationships among arithmetic concepts, rather than the relationships among sample data.

#### 4.3.3. The More Efficient Evidence Concepts Given, the Better the Inference Results

For the inference results in cases 4, 5, and 7 in Table 6, given two evidence concepts “multiples of 4 ( $E=1$ )” and “ends in 4 ( $N=1$ )”, we can infer that the probabilities of “even ( $B=1$ )” and “multiples of 8 ( $I=1$ )” are 1 and 0.43659, respectively. We conclude that the probability of “multiples of 8 ( $I=1$ )” is low. Next, we add an evidence concept “powers of 2 ( $T=1$ )”, and the probability of belonging to the concept of “multiples of 8 ( $I=1$ )” is increased to 0.833333. Thus, we have that the increase in evidence concepts makes concept inference more certain in this case. Furthermore, we continue to add two evidences “not odd ( $A=0$ )” and “not multiples of 7 ( $H=0$ )”; this is case 7 in Table 6, but the probability of “multiples of 8 ( $I=1$ )” does not increase again. This shows that the additional two evidence does not have much effect on the inference result, and they are inefficient evidence. Therefore, it can be concluded that both the efficiency and the number of evidence concepts affect the inference results. The more effective evidence given, the better the inference results, which makes the uncertainty decrease faster.

## 5. Conclusions

To learn and infer arithmetic concepts from positive integer numbers, we have addressed an arithmetic concept Bayesian network (ACBN) model and mainly studied the ACBN model learning algorithm, evidence concept learning method, and ACBN inference algorithm.

The method proposed in this paper also raises other research issues. The BN model has strong abilities of knowledge representation and probabilistic inference, but in this paper, we only discuss the arithmetic concept BN model and its learning and inference algorithms in “number game”. There is still much work that needs to be done to solve significant problems in practice, such as image recognition by stimulating human level learning and inference abilities.

## Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (No. 61432011, U1435212, 61322211, 61672332), the Postdoctoral Science Foundation of China (No. 2016M591409), and the Natural Science Foundation of Shanxi Province, China (No. 201801D121115, 2013011016-4, 2014011022-2).

## References

1. J. Pearl, “Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference,” Morgan Kaufmann Publishers, San Mateo, California, 1988
2. L. W. Zhang and H. P. Guo, “Introduction to Bayesian Networks,” Science Press, 2006
3. Y. W. Park and D. Klabjan, “Bayesian Network Learning via Topological Order,” *Journal of Machine Learning Research*, Vol. 18, pp. 1-32, 2017
4. M. Scanagatta, G. Corani, C. P. de Campos, and M. Zaffalon, “Approximate Structure Learning for Large Bayesian Networks,” *Machine Learning*, Vol. 107, pp. 1209-1227, 2018
5. R. Mateescu, K. Kask, V. Gogate, and R. Dechter, “Join-Graph Propagation Algorithms,” *Journal of Artificial Intelligence Research*, Vol. 37, pp. 279-328, 2010
6. C. J. Butz, J. S. Oliveira, A. E. D. Santos, and A. L. Madsen, “Inference with Simple Propagation,” in *Proceedings of JMLR Workshop and Conference Proceedings*, Vol. 52, pp. 62-73, 2016
7. B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, “Human-Level Concept Learning Through Probabilistic Program Induction,” *Science*, Vol. 350, No. 6266, pp. 1332-1338, 2015
8. D. George, W. Lehrach, K. Kinsky, M. Lázaro-Gredilla, C. Laan, B. Marthi, et al., “A Generative Vision Model that Trains with High Data Efficiency and Breaks Text-based CAPTCHAS,” *Science*, Vol. 358, No. 6368, 2017
9. X. Ma, T. Zhao, R. S. Wen, Z. J. Wu, and Q. Wang, “Motion Recognition based on Concept Learning,” in *Proceedings of IEEE International Conference on Instrumentation and Measurement Technology (I2MTC)*, pp. 1-6, 2017
10. K. P. Murphy, “Machine Learning: A Probabilistic Perspective,” MIT Press, 2012
11. D. Koller and N. Friedman, “Probabilistic Graphical Models: Principles and Techniques,” MIT Press, 2009
12. H. Lähdesmäki and I. Shmulevich, “Learning the Structure of Dynamic Bayesian Networks from Time Series and Steady State

Measurements,” *Machine Learning*, Vol. 71, pp. 185-217, 2008

13. C. P. de Camposab, M. Scanagatta, G. Corani, and M. Zaffalon, “Entropy-based Pruning for Learning Bayesian Networks Using BIC,” *Artificial Intelligence*, Vol. 260, pp. 42-50, 2018
14. M. Scutari, “Learning Bayesian Networks with the Bnlearn R Package,” *Journal of Statistical Software*, Vol. 35, No. 3, pp. 1-22, 2010

**Yali Lv** is an associate professor at Shanxi University of Finance & Economics of China. She received her Ph.D. from Tianjin University. Her research interests include probabilistic reasoning, concept learning, data mining, and machine learning.

**Tong Jing** is a master's candidate at Shanxi University of Finance & Economics of China. His research interests include Bayesian machine learning and concept learning.

**Yuhua Qian** is a professor in the Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education at Shanxi University of China. He received his Ph.D. from Shanxi University. His research interests include granular computing, social computing, and machine learning for big data.

**Jiye Liang** is a professor in the Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education at Shanxi University of China. He received his Ph.D. from Xi'an Jiaotong University. His research interests include computational intelligence, granular computing, data mining, and knowledge discovery.

**Jianai Wu** is a master's candidate at Shanxi University of Finance & Economics of China. Her research interests include data mining and financial data analysis.

**Junzhong Miao** is a master's candidate at Shanxi University of Finance & Economics of China. His research interests include Bayesian machine learning and concept learning.